ELEC-204 HW-4

Honor Code :

"I hereby certify that I have completed this assignment on my own without any help from anyone else.

I understand that the only sources of authorized information in this open-book exam are
(i) the course textbook, and
(ii) the lecture notes self-taken or distributed by the instructor at Blackboard for this class.

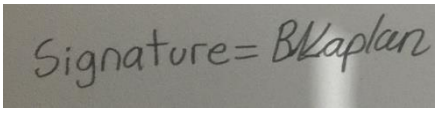
I have not used, accessed, received or distributed any information from/to any other unauthorized source in taking this exam.

The effort in the assignment thus belongs completely to me."

Name-Surname: Barış KAPLAN

ID Number: 69054

E-mail address: bkaplan18@ku.edu.tr



SOLUTIONS OF THE HOMEWORK PROBLEMS

1-) At the beginning of the parallel adder, there are three inputs called A, B, and Cin. At the beginning; the first Cin is equal to 0, A is equal to 1, and B is equal to 0. Since the first Cin is equal to 0, Sum-1 is equal to the result of A xor B. Since "1 xor 0" results in 1, the Sum-1 is equal to 1. There is no overflow for the result of the addition of A and B here because this result is smaller than 2(in decimal, 2 is 10 in binary). So, the first Cout is equal to 0. The first Cout becomes the second Cin. So, the second Cin is equal to 0. Then, we add the second digits. To add them, we apply the xor operation to these digits. Sum-2 is equal to the result of "1 xor 1". So, Sum-2 is equal to 0. There is an overflow for the result of the addition of A and B here because this result is not smaller than 2 (in decimal, 2 is 10 in binary). Therefore, the second Cout is equal to 1. The second Cout becomes the third Cin. So, the third Cin is equal to 1. Finally, we should apply the xor operation to the third digits of the binary numbers. Here, there is a carry bit which comes from the previous addition. So, we should apply the xor operation to A,

B, and Cin. “(1 xor 0) xor 1” results in 0. So, the Sum-3 is equal to 0. There is an overflow here. Because the sum of 1, 0, and 1 is not smaller than 2 (in decimal). The last Cout is equal to 1, and Sum-4 is equal to 1.

1+1= 01 (In Binary)
1: 1-digit binary number
1: 1-digit binary number
10: 2-digit binary number (We need more digits to represent the sum in binary, so we say that there is an overflow here.)

Sum-1: 1
Sum-2: 0
Sum-3: 0
Sum-4: 1

So, the result of the addition of 111 and 010 is 1001.

111+010= 1001

//Add the first digits(1+0=1, carry does not exist). Add the second digits(1+1=0, carry exists). Add the third digits(1+0= 1 -> carry comes from the previous addition-> 1+1= 0-> the result of the addition is 0-> carry exists). So, 1001 is the result of the addition of 111 and 010.

2-)

A	B	C	Result
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

For the first result and second result-> F=0

For the third result and fourth result-> F=C’

For the fifth result and sixth result -> F= C

For the last two results -> F=1

Note: You can find the design of the 4 to 1 multiplexer in the attached images to my submission.

3-)

The sequence of inputs for A0: 10001
The sequence of inputs for A1: 00110
The sequence of inputs for A2: 11100

Y= A0'*A1*A2+A1'*A0*A2+A0'*A1*A2' (The boolean expression version of the decoding logic in the given figure.)

For A0=1, A1=0, A2=1
A0'= 0
A1'= 1
A2'= 0

A0= 1
A1= 0
A2= 1

A0'*A1*A2= 0
A1'*A0*A2= 1
A0'*A1*A2'= 0
Y= 0+1+0= 1

For A0=0, A1=0, A2=1
A0'= 1
A1'= 1
A2'= 0

A0= 0
A1= 0
A2= 1

A0'*A1*A2= 0
A1'*A0*A2= 0
A0'*A1*A2'= 0
Y= 0+0+0= 0

For A0=0, A1=1, A2=1

A0= 0
A1= 1
A2= 1

A0'= 1
A1'= 0
A2'= 0

A0'*A1*A2= 1
A1'*A0*A2= 0
A0'*A1*A2'= 0
Y= 1+0+0= 1

For A0= 0, A1= 1, A2= 0
A0= 0
A1= 1
A2= 0

A0'= 1
A1'= 0
A2'= 1

A0'*A1*A2= 0
A1'*A0*A2= 0
A0'*A1*A2'= 1
Y= 0+0+1= 1

For A0= 1, A1= 0, A2= 0
A0= 1
A1= 0
A2= 0

A0'= 0
A1'= 1
A2'= 1

A0'*A1*A2= 0
A1'*A0*A2= 0
A0'*A1*A2'= 0
Y= 0+0+0= 0

So, the sequence of inputs for Y: 10110

Note: You can find the output waveform in the attached images to my submission.